

# timerate manual page - Built-In Commands

 [tcl-lang.org/man/tcl/TclCmd/timerate.htm](http://tcl-lang.org/man/tcl/TclCmd/timerate.htm)

## NAME

timerate — Calibrated performance measurements of script execution time

## SYNOPSIS

**timerate** *script* ?*time*? ?*max-count*?

**timerate** ?-**direct**? ?-**overhead** *double*? *script* ?*time*? ?*max-count*?

**timerate** ?-**calibrate**? ?-**direct**? *script* ?*time*? ?*max-count*?

## DESCRIPTION

The **timerate** command does calibrated performance measurement of a Tcl command or script, *script*. The *script* should be written so that it can be executed multiple times during the performance measurement process. Time is measured in elapsed time using the finest timer resolution as possible, not CPU time; if *script* interacts with the OS, the cost of that interaction is included. This command may be used to provide information as to how well a script or Tcl command is performing, and can help determine bottlenecks and fine-tune application performance.

The first and second form will evaluate *script* until the interval *time* given in milliseconds elapses, or for 1000 milliseconds (1 second) if *time* is not specified.

The parameter *max-count* could additionally impose a further restriction by the maximal number of iterations to evaluate the script. If *max-count* is specified, the evaluation will stop either this count of iterations is reached or the time is exceeded.

It will then return a canonical tcl-list of the form:

```
0.095977 µs/# 52095836 # 10419167 #/sec 5000.000 net-ms
```

which indicates:

- the average amount of time required per iteration, in microseconds ([[lindex](#) \$result 0])
- the count how many times it was executed ([[lindex](#) \$result 2])
- the estimated rate per second ([[lindex](#) \$result 4])
- the estimated real execution time without measurement overhead ([[lindex](#) \$result 6])

The following options may be supplied to the **timerate** command:

### **-calibrate**

To measure very fast scripts as exactly as possible, a calibration process may be required. The **-calibrate** option is used to calibrate **timerate** itself, calculating the

estimated overhead of the given script as the default overhead for future invocations of the **timerate** command. If the *time* parameter is not specified, the calibrate procedure runs for up to 10 seconds.

Note that calibration is not thread safe in the current implementation.

### **-overhead double**

The **-overhead** parameter supplies an estimate (in microseconds) of the measurement overhead of each iteration of the tested script. This quantity will be subtracted from the measured time prior to reporting results. This can be useful for removing the cost of interpreter state reset commands from the script being measured.

### **-direct**

The **-direct** option causes direct execution of the supplied script, without compilation, in a manner similar to the **time** command. It can be used to measure the cost of **Tcl\_EvalObjEx**, of the invocation of canonical lists, and of the uncompiled versions of bytecoded commands.

As opposed to the **time** command, which runs the tested script for a fixed number of iterations, the **timerate** command runs it for a fixed time. Additionally, the compiled variant of the script will be used during the entire measurement, as if the script were part of a compiled procedure, if the **-direct** option is not specified. The fixed time period and possibility of compilation allow for more precise results and prevent very long execution times by slow scripts, making it practical for measuring scripts with highly uncertain execution times.

## **EXAMPLES**

---

Estimate how fast it takes for a simple Tcl **for** loop (including operations on variable *i*) to count to ten:

```
# calibrate
timerate -calibrate {}

# measure
timerate { for {set i 0} {$i<10} {incr i} {} } 5000
```

Estimate how fast it takes for a simple Tcl **for** loop, ignoring the overhead of the management of the variable that controls the loop:

```
# calibrate for overhead of variable operations
set i 0; timerate -calibrate {expr {$i<10}; incr i} 1000

# measure
timerate {
    for {set i 0} {$i<10} {incr i} {}
} 5000
```

Estimate the speed of calculating the hour of the day using **clock format** only, ignoring overhead of the portion of the script that prepares the time for it to calculate:

```
# calibrate
timerate -calibrate {}

# estimate overhead
set tm 0
set ovh [lindex [timerate {
    incr tm [expr {24*60*60}]
}] 0]

# measure using estimated overhead
set tm 0
timerate -overhead $ovh {
    clock format $tm -format %H
    incr tm [expr {24*60*60}]; # overhead for this is ignored
} 5000
```

# time manual page - Built-In Commands

---

 [tcl-lang.org/man/tcl/TclCmd/time.htm](http://tcl-lang.org/man/tcl/TclCmd/time.htm)

## **NAME**

---

time — Time the execution of a script

## **SYNOPSIS**

---

**time** *script* *?count?*

## **DESCRIPTION**

---

This command will call the Tcl interpreter *count* times to evaluate *script* (or once if *count* is not specified). It will then return a string of the form

**503.2 microseconds per iteration**

which indicates the average amount of time required per iteration, in microseconds. Time is measured in elapsed time, not CPU time.

## **EXAMPLE**

---

Estimate how long it takes for a simple Tcl **for** loop to count to a thousand:

```
time {  
    for {set i 0} {$i<1000} {incr i} {  
        # empty body  
    }  
}
```